SAND2015-0537C

Domain Decomposition Solver Strategies for Future Platforms

Clark Dohrmann

Computational Solid Mechanics &
Structural Dynamics Department

ACS Meeting
Sandia National Laboratories
February 2-5, 2015

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

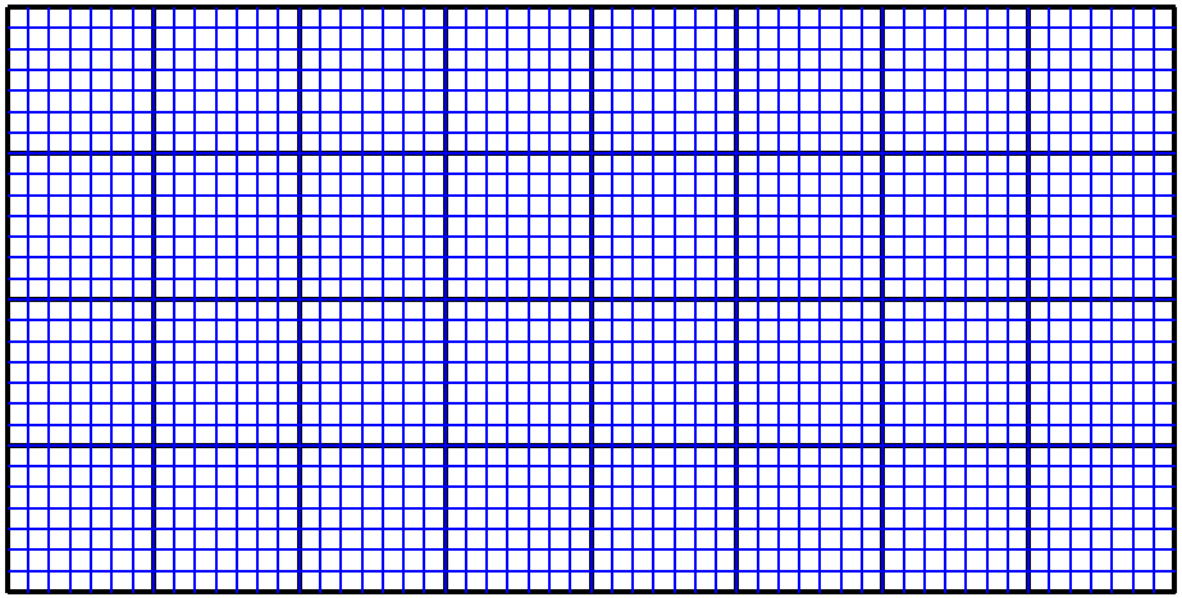
Outline

- **Domain Decomposition:**
  - Additive Schwarz Preconditioners
  - Computational Kernels
  - Flexibility
- **Parallelism (Trinity):**
  - Vectorization
  - Threads
  - MPI
- **Current Efforts:**
  - Threaded sparse direct solvers
  - Parallel constraint elimination
  - On the fly re-decomposition

Domain Decomposition

Sandia National Laboratories

Two-level Additive Schwarz Preconditioner:



$$Ax = b$$

$$AM^{-1}y = b$$

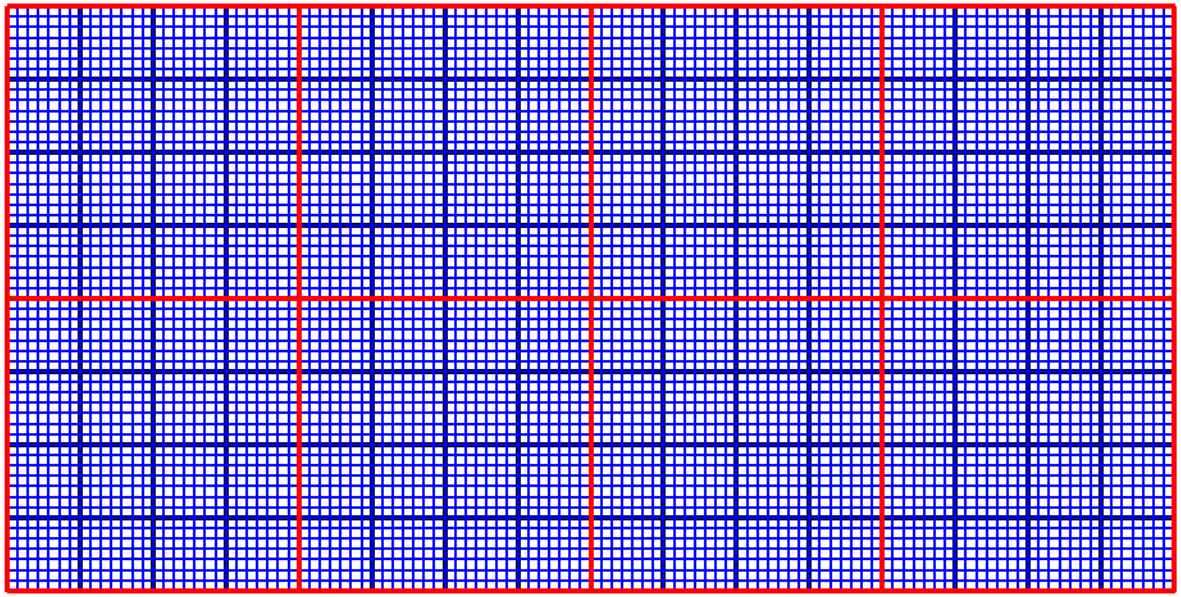
$$M^{-1}r = \sum_{i=1}^{N} R_i^T (R_i A R_i^T)^{-1} R_i r + \Phi(\Phi^T A \Phi)^{-1}\Phi^T r$$

$R_i =$ Boolean matrix $\qquad \Phi =$ interpolation matrix

Domain Decomposition

Sandia National Laboratories

Multi-level Additive Schwarz Preconditioner:



$$Ax = b$$

$$AM^{-1}y = b$$

$$M^{-1}r = \sum_{j=1}^{M-1} \sum_{i=1}^{N_j} R_{ij}^T (R_{ij} A_j R_{ij}^T)^{-1} R_{ij} r_j + \Phi_M (\Phi_M^T A \Phi_M)^{-1} \Phi_M^T r$$

$$r_j = \Phi_j^T r, \quad \Phi_1 = I \qquad A_j = \Phi_j A \Phi_j^T$$

Domain Decomposition

Sandia National Laboratories

- **Computational Kernels:**
  - **Sparse matrix-vector multiplication**
    - **Application of operator/coarse interpolations**
    - **Tpetra/Kokkos**
  - **Solution of linear systems**
    - **Threaded/accelerated factorizations and solves**
      - **MKL Pardiso**
      - **Sandia efforts (Trilinos)**
    - **Inexact subdomain solves**
      - **Reduced memory, manycore**
  - **Dense linear algebra**
    - **Iterative solution acceleration**
      - **Subspace recycling (projections)**
    - **Sparse direct solvers (supernodal variants)**
  - **Constraint equations (more later)**

Domain Decomposition

Sandia National Laboratories

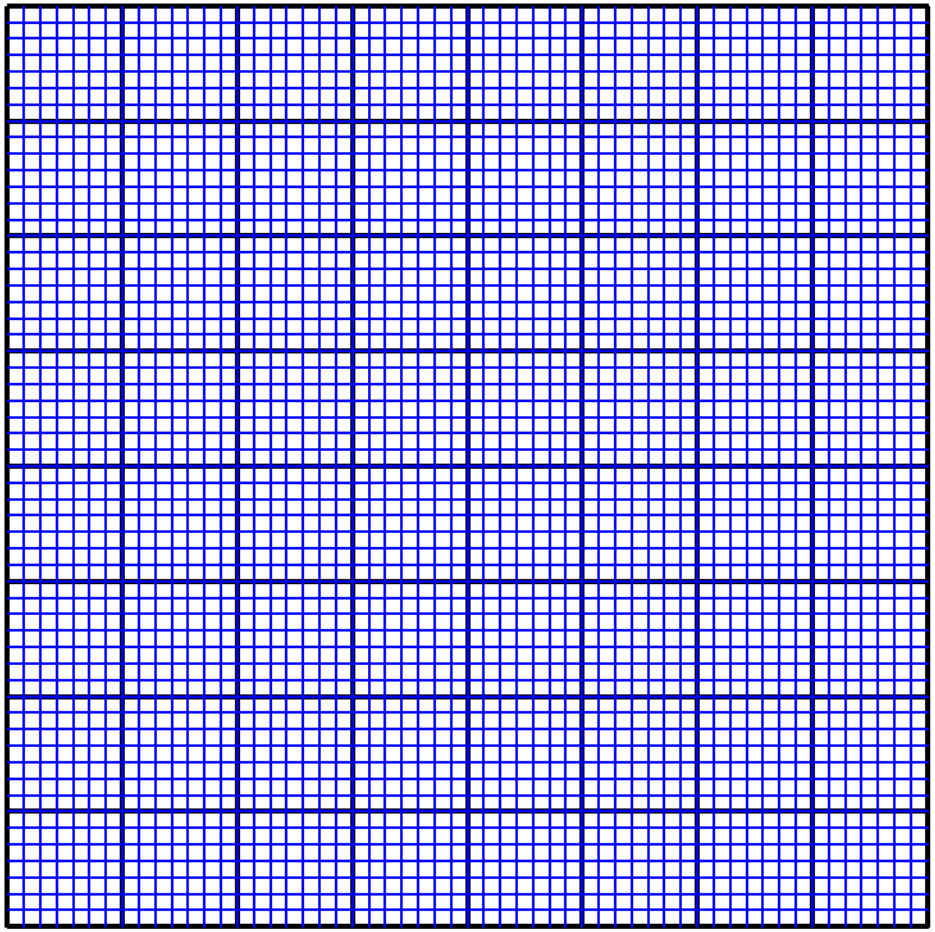
- **Flexibility:**
  - **Historically one core/MPI process and one core/subdomain**
  - **Several options**
    - **Threaded linear solves**
      - **Keeps threads busy while not introducing additional subdomains**
    - **Multiple subdomains per core (over-decomposition*)**
      - **To help address load imbalance**
      - **More subdomains, but helps keep cores busy**
      - **Viewed as introducing more subdomains or inexact local solver**
    - **Multiple cores per MPI process**
      - **e.g. one MPI process per Trinity quadrant**
      - **Each MPI process involves multiple subdomains**
      - **Can be viewed as inexact local solver for larger subdomain**
  - **Examples**

*Richard Barrett, Mike Glass suggestion

Domain Decomposition

Sandia National Laboratories

- **Flexibility: Example 1, business as usual**



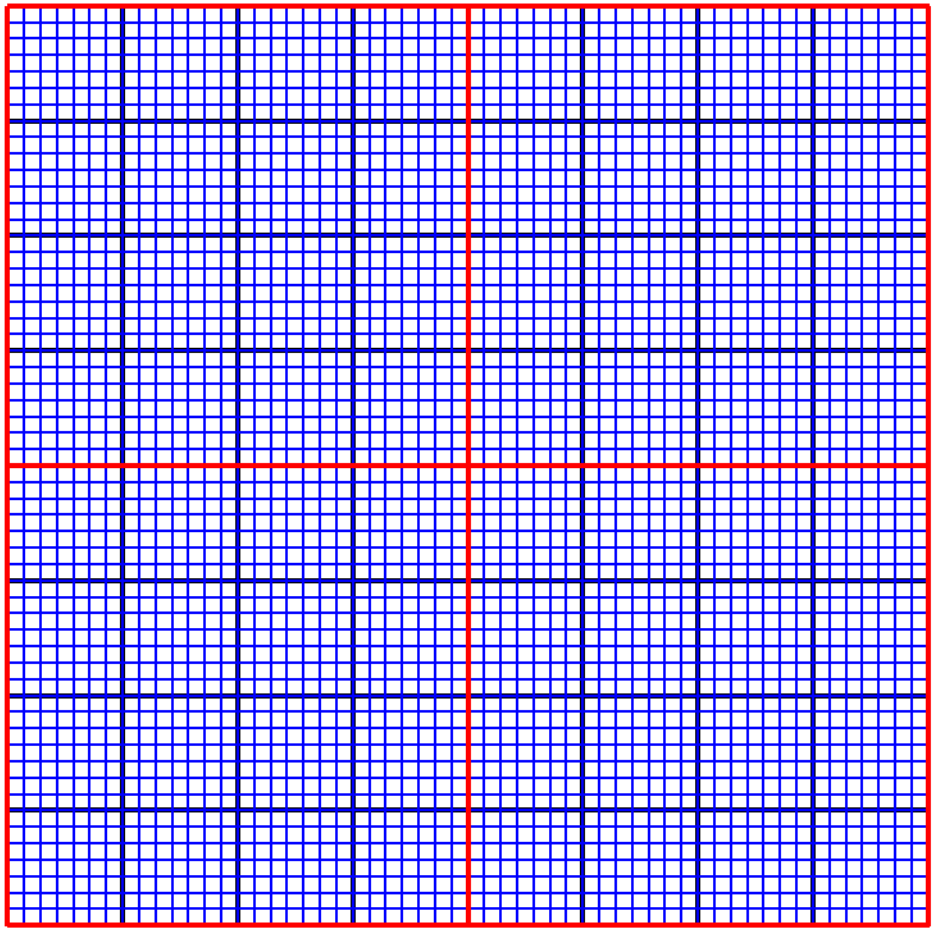
- 64 cores
- 1 core per subdomain
- 1 core per MPI process

view of single node

Domain Decomposition

Sandia National Laboratories

- **Flexibility: Example 2 (fewer MPI procs than cores)**

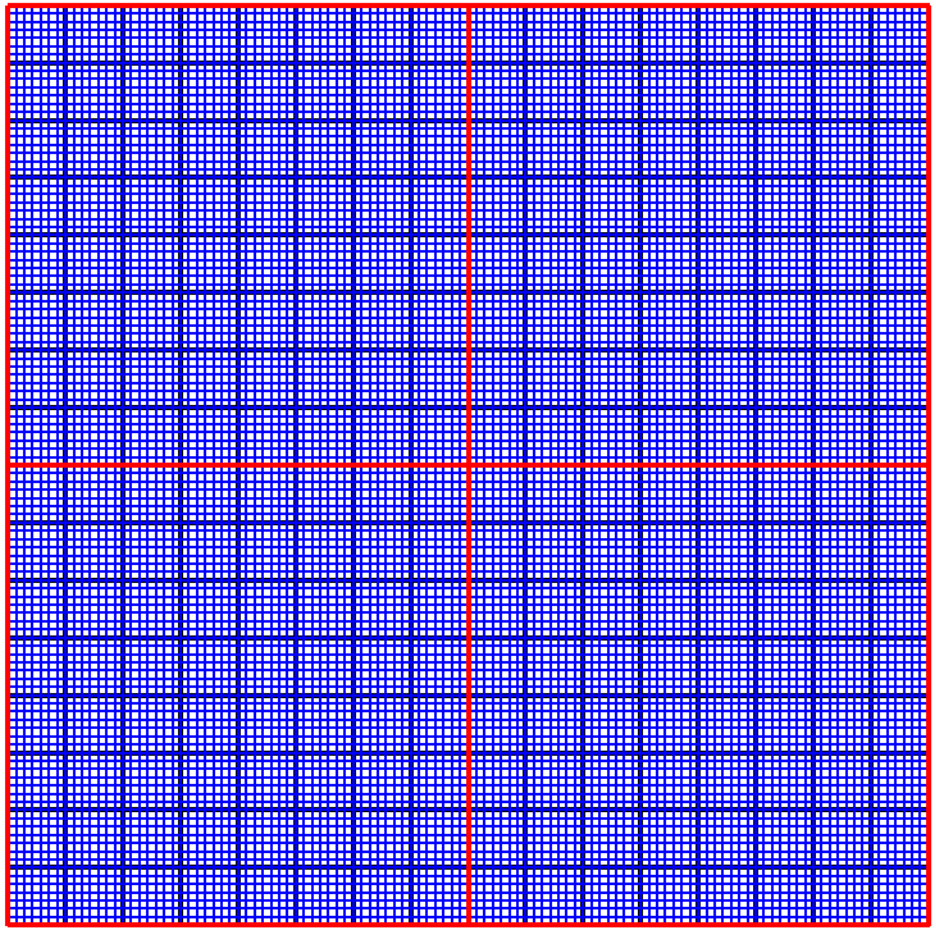


- 64 cores
- 16 cores per larger subdomain
- 1 smaller subdomain for each core
- 16 cores per MPI process
- “local” solves for larger subdomain can be done inexactly
- e.g. larger subdomain associated with quadrant

view of single node

Domain Decomposition

- **Flexibility: Example 3 (over-decomposition)**



- 64 cores
- 16 cores per larger subdomain
- 4 smaller subdomains for each core
- 16 cores per MPI process
- “local” solves for larger subdomain can be done inexactly
- e.g. larger subdomain associated with quadrant

view of single node

Parallelism (Trinity Focus)

Sandia National Laboratories

- **Vectorization:**
  - Dense linear algebra
  - Rely on vendor BLAS
  - For sparse linear solves & projections
- **Threads:**
  - Direct solvers
    - MKL/Pardiso & Sandia efforts so far
  - Over-decomposition task parallelism
- **MPI:**
  - Multilevel Preconditioner
    - Tpetra-based
    - Energy-minimizing (algebraic) coarse spaces
    - Concurrency possible at all levels for additive variant

Current Efforts

Sandia National Laboratories

- **Threaded Sparse Direct Solvers:**
  - **MKL/Pardiso**
  - **Sandia efforts**
    - **Cholesky, LDL^T, LU**
    - **Threaded factorizations & triangular solve**
    - **Different threading approaches**
    - **Future: inexact manycore**
    - **To be accessible via Trilinos**
  - **Can swap in alternatives as become available**

Current Efforts

- **Parallel Constraint Elimination:**

$$\begin{bmatrix} A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

$$Cx = 0 \Rightarrow x = Fx_i \Rightarrow F^T AFx_i = F^T b$$

  - **Sparse LU for rectangular systems**
    - **Constraint matrix C may be singular**
    - **Need to identify redundant constraints**
  - **Great for certain types of constraint equations**
    - **Connecting dissimilar meshes**
    - **Enforcing rigid constraints**
  - **Constraint-aware decompositions can be important**

Current Efforts

Sandia National Laboratories

- **On the fly re-decomposition:**
  - **Good load balance very important**
  - **Ignoring constraints or physics can cause problems**
    - **Poorly shaped subdomains**
    - **Poor load balance (e.g. structural-acoustics)**
  - **Needed for over-decomposition**
  - **Vertex separators well-suited for domain decomposition**
    - **Keep interface of decomposition smaller**
    - **Follows naturally from element decomposition**
    - **Not so easy if element information lost (e.g. matrix only)**
  - **Algorithm:**
    - **Based on hypergraph model where vertices are non-zero entries of matrix and hyperedges are rows/columns of matrix**
    - **Implementation based on Zoltan/phg**

Closing Remarks

Sandia National Laboratories

- **Domain Decomposition Solvers**
  - **Natural fit for future platforms**
    - **Flexibility to optimize performance**
    - **Lots of dense linear algebra**
  - **Can benefit greatly from Trilinos-related efforts**
    - **Sparse triangular factorizations and triangular solves**
    - **Sparse matrix-vector multiplication**
    - **Isolation from hardware specifics**